Computers and Mathematics with Applications 62 (2011) 392–400

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

A symmetric rank-one method based on extra updating techniques for unconstrained optimization

Farzin Modarres [a,*], Malik Abu Hassan [b], Wah June Leong [b]

[a] Department of Mathematics, Science and Research Branch, Islamic Azad University, Tabriz, East Azerbaijan, Iran

[b] Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

ARTICLE INFO

Article history:
Received 1 July 2010
Received in revised form 12 May 2011
Accepted 12 May 2011

Keywords:
Unconstrained optimization
Hessian approximation
Symmetric rank-one update
Extra updating techniques

ABSTRACT

In this paper, we present a new symmetric rank-one (SR1) method for the solution of unconstrained optimization problems. The proposed method involves an algorithm in which the usual SR1 Hessian is updated a number of times in a way to be specified in some iterations, to improve the performance of the Hessian approximation.

In particular, we discuss how to consider a criterion for indicating at each iteration whether it is necessary to employ extra updates. However it is well known that there are some theoretical difficulties when applying the SR1 update. Even for a current positive definite Hessian approximation, it is possible that the SR1 update may not be defined or the SR1 update may not preserve positive definiteness at some iterations. We then employ a restarting procedure that guarantees that updated matrices will be well-defined while preserving positive definiteness of updates. Numerical results support these theoretical considerations. They show that the implementation of the SR1 method using extra updating techniques improves the performance of the SR1 method substantially for a number of test problems from the literature.

1. Introduction

Quasi-Newton (QN) methods have been applied efficiently to solve a wide range of unconstrained optimization problems given by

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable. Among the existing QN methods the symmetric rank-one (SR1) method is one of the most efficient methods for solving this problem. At the kth iteration, assume that x_k is the vector of variables such that $f(x_k)$ is the least calculated value of the objective function so far. If $g_k \neq 0$, then we use the following quadratic model for approximating f around x_k:

$$Q_k(x_k + d) = \frac{1}{2} d^T B_k d + g_k^T d + f(x_k), \tag{2}$$

where g_k denotes the current gradient of f at x_k, $d \in \mathbb{R}^n$ and B_k is an $n \times n$ symmetric matrix that approximates the true Hessian G_k at x_k. By using either line-search or trust-region frameworks, a new iterate point x_{k+1} is obtained such that

$$f(x_{k+1}) < f(x_k). \tag{3}$$

* Corresponding author. Tel.: +60 173846343; fax: +60 389437958.
E-mail addresses: farzin.modarres@yahoo.ca, farzin@math.upm.edu.my (F. Modarres).

0898-1221/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2011.05.022

For the trust-region method, the radius may be adjusted several times to realize (3). For the line-search method, condition (3) can be realized by some line-search strategy such as using the Wolfe conditions or the Armijo condition.

Our attention in this paper will be on the line-search method. Then for a given initial symmetric and positive definite B_1, we may update B_k to a new Hessian approximation B_{k+1} by using the SR1 formula. The class of line-search methods starts with an initial approximation, x_1, of a solution, x_*, and generates a new approximation by means of the following iterative form:

$$x_{k+1} = x_k + \alpha_k d_k; \quad k \geq 1, \tag{4}$$

where α_k is a step length parameter and d_k is the search direction found by solving the set of equations

$$B_k d_k = -g_k. \tag{5}$$

Note that positive values of the step length parameter, which ensure sufficient reductions in f, exist when the descent property $g_k^T d_k < 0$ is satisfied; such a condition is guaranteed if the Hessian approximation B_k is positive definite. In the class of quasi-Newton methods, this matrix is updated in terms of the vector pair

$$s_k = x_{k+1} - x_k, \tag{6}$$

and

$$y_k = g_{k+1} - g_k, \tag{7}$$

to a new Hessian approximation B_{k+1} such that the quasi-Newton condition

$$B_{k+1} s_k = y_k, \tag{8}$$

holds (for further details, see e.g. [1]).

In particular, we suppose that

$$B_{k+1} = \mathrm{SR1}(B_k, s_k, y_k), \tag{9}$$

where, for any B, s and y,

$$\mathrm{SR1}(B, s, y) = B + \frac{(y - Bs)(y - Bs)^T}{s^T (y - Bs)}, \tag{10}$$

is the SR1 updating formula. Investigations on the SR1 update indicate that SR1 can outperform the well known BFGS update

$$\mathrm{BFGS}(B, s, y) = B - \frac{Bss^T B}{s^T Bs} + \frac{yy^T}{y^T s}, \tag{11}$$

in the context of optimization, where either the approximate Hessian can be expected to be positive definite or a trust-region framework is used [2–5]. The SR1 update makes a rank-one change to the previous Hessian approximation B_k; hence it is evidently simple in form. The SR1's hereditary property on (convex) quadratics implies that if n (consecutive) linearly independent iterates are taken, then the Hessian approximation converges to the true (quadratic) Hessian in at most n steps and hence the solution can be reached in at most $n + 1$ steps (if the update exists at each step). Moreover SR1 update (10) is also the unique symmetric rank-one update that satisfies the quasi-Newton equation (8). Self-complementarity of the SR1 update is another attractive property, i.e. given any update formula for B_{k+1} that depends only on B_k, s, and y we can obtain an update

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{y_k^T (s_k - H_k y_k)} \tag{12}$$

that satisfies the QN equation for the inverse Hessian approximation

$$H_{k+1} y_k = s_k. \tag{13}$$

This so-called complementarity operation on the SR1 update B_{k+1} yields the rank-one update. From the uniqueness of this update, we conclude immediately that the inverse of the update is the same as its complement (obviously the inverse of the update will satisfy the QN equation (13)). Although the SR1 update possesses desirable features, it has some major drawbacks. It does not necessarily generate a positive definite matrix even if the current update is positive definite, and the denominator in the SR1 update may become zero. As regards the drawbacks of the SR1 update, some safeguards may prevent the possible breakdown and the presence of numerical instabilities. The need of maintaining positive definiteness for the approximation to the Hessian matrix of the objective function has encouraged a lot of researchers to seek some modified schemes of update which possess not only good features of original SR1 update but also greater numerical stability.

To overcome the drawbacks of the SR1 update, various variants of the SR1 method have been proposed, for example, a self-adjusting variable metric algorithm, a switching algorithm, a multi-direction parallel algorithm, and sizing the current update and restart procedure (see, for instance, [6–12]). Osborne and Sun [7] introduced a scaling parameter in the SR1

update to ensure that successive estimates of the inverse Hessian are positive definite, whereas Dennis and Wolkowicz [10] and Khalfan [2] sized the approximations of the Hessian using a sizing factor to maintain positive definite updates.

Conn et al. [4] proved that the sequence of matrices generated by the SR1 updates converges to the actual Hessian, provided that the sequence of steps taken, x_k, is uniformly linearly independent. Using this result it is not difficult to prove that the rate of convergence is q-superlinear. On the other hand, Khalfan et al. [3] presented a new analysis that shows that the SR1 update with a line search is $n+1$-step q-superlinearly convergent without the assumption of linearly independent iterates. In particular, the computational results show that the SR1 update is very competitive with the widely used BFGS update.

The efficiency of the QN methods depends strongly on the quality of the Hessian approximations, B_k. Though the BFGS method is robust in correcting the eigenvalues of the Hessian approximation B_k, this correction is found to be inadequate in practice when the eigenvalues of B_k are large (see, e.g., [13,14]). To overcome this difficulty, several modification techniques have been introduced into the Hessian approximations to be applied before updating by the BFGS formula (e.g., see [15,16]). Al-Baali [17] updated the usual BFGS Hessian a number of times, depending on the information of the first-order derivatives, to obtain a new Hessian approximation at each iteration. Since extra updates seem to improve the poor quality of the Hessian approximation, such consideration motivate us to consider the possibility of employing these updates in the SR1 method, where extra updates of the Hessian are made in a way to be specified at some iterations.

Moreover in order to overcome the difficulties of using the SR1 formula we employ a restart procedure for the SR1 update. The restart procedure provides a replacement for the non-positive definite or unbounded (indefinite) B_k with a scaling of the identity matrix. Hence the scaling factor is incorporated whenever these difficulties arise for the SR1 update.

In Section 2, we will describe the extra updates scheme and derive the resulting extra updates method for the SR1 formula. In Section 3, we consider a decisive factor for specifying at each iteration whether it is necessary to employ extra updates. In Section 4, we address the question of maintaining the positive definite updates in the approximations, while Section 5 proposes an extra SR1 algorithm which attempts to combine the best features of the SR1 update and extra updates schemes. The results of numerical experiments using the proposed method are described in Section 6.

Throughout the paper $\|.\|$ will be the Euclidean norm, although in some cases it can be replaced by an arbitrary norm.

2. The symmetric rank-one method using the extra updates scheme

In this section, we briefly describe the properties of the extra update method.

Consider the following extra updating technique for some $m \leq n$ (n is the dimension of the problem):

$$\nabla^2 f(x_{k+1})\rho_l = \zeta_l, \quad 1 \leq l \leq m, \tag{14}$$

where $\{\rho_l\}$ and $\{\zeta_l\}$ are any sequences of independent vectors. Since the true Hessian is not available in practice, we can use some finite differences to approximate the left-hand side of (14). Thus generally the above equation becomes

$$\nabla^2 f(x_{k+1})\rho_l \approx \zeta_l, \quad 1 \leq l \leq m. \tag{15}$$

Note that since generally finite differences prove expensive, the above method of extra updates is usually intended for implementation on a parallel machine with m processors.

Al-Baali [18] conjectured that curvature information where the vectors $\{\rho_l\}$ and $\{\zeta_l\}$ satisfy

$$\rho_l = s_{k-m+l}, \qquad \zeta_l = y_{k-m+l}, \quad 1 \leq l \leq m, \tag{16}$$

may be employed m times ($m > 1$) to improve the BFGS Hessian approximations. In fact, using (6)–(7) and the Taylor series expansion, we see that the choice (16) implies approximations like (15) with k replaced by $k-m+l$. In fact, one can observe through (6)–(7) and the Taylor series that the choice (16) implies approximations similar to (15) where k has been replaced by $k-m+l$. Let the current SR1 update B_k be given as follows: we proceed at each iteration k by carrying out a single-step quasi-Newton update as follows:

$$\bar{B}_{k+1}^{(l)} = \mathrm{SR1}(\bar{B}_k, s_k, y_k). \tag{17}$$

The updated Hessian approximation is used in the subsequent $m-1$ sub-iterations for updating the SR1 Hessian a number of times as follows:

$$\bar{B}_{k+1}^{(l+1)} = \mathrm{SR1}(\bar{B}_{k+1}^{(l)}, \rho_l, \zeta_l), \quad l = 1, 2, \ldots, m, \tag{18}$$

where m is a prescribed constant and the vectors $\{\rho_l\}$ and $\{\zeta_l\}$ satisfy (16). The $(m+1)$th update is used to define a new Hessian approximation, i.e.

$$B_{k+1} = \bar{B}_{k+1}^{(m+1)}. \tag{19}$$

Hence the relation (17) employs an extra m SR1 updates in terms of the vector pairs

$$\{s_{k-m+l}, y_{k-m+l}\}, \quad 1 \leq l \leq m, \tag{20}$$

to obtain the updated Hessian approximation (19). This Hessian (referred to as the EXSR1 Hessian) reduces to the SR1 Hessian when $m \leq 1$. Note that k replaces m whenever $k \leq m$. But once $k > m$, the oldest vector pair is deleted from the sequence (16) and is replaced by the pair (6) and (7).

Note that for $m = 1$, the formula for extra updates (18) with the starting matrix (17) produces the same SR1 Hessian. Thus, the choice $m = 1$ (like $m = 0$) does not provide a new Hessian approximation and should be avoided. Several properties of this extra updating scheme are given in [17]. For the purpose of numerical illustration, we will use $m = 2$ in this paper.

3. A criterion for employing extra updates

It remains to deal with the issue of a criterion for employing extra updates to the SR1 method. From the numerical results of Al-Baali [17], we observe that because at some iterations extra updates might be redundant or reduce the quality of this Hessian, extra updates should be employed only when they satisfy certain properties. In fact, our main reason for adopting an extra update technique for SR1 is to encourage a more rapid reduction in the eigenvalue of B_k. To do this, it is important to make the choice on the basis of an estimate of the determinant of the updated Hessian approximation. But since the eigenvalues are not computed in practice and the curvature can be written as

$$s_k^T y_k = s_k^T \bar{G}_k s_k,$$

where

$$\bar{G}_k = \int_0^1 G(x_k + t s_k) \mathrm{d}t$$

is the average Hessian matrix along s_k, we can estimate their size on the basis of the scalars

$$b_k = \frac{s_k^T B_k s_k}{s_k^T y_k}, \tag{21}$$

and

$$h_k = \frac{y_k^T H_k y_k}{s_k^T y_k}, \tag{22}$$

by examining the determinant of the SR1 Hessian

$$\det(B_{k+1}^{\mathrm{SR1}}) = -\left(\frac{1 - h_k}{1 - b_k}\right) \det(B_k). \tag{23}$$

Eq. (23) can be found in [19] which follows from (9) and (10) and

$$\det(I + uv^T) = 1 + u^T v. \tag{24}$$

where I is the identity matrix, and u and v are any vectors.

In fact, we seek further correction to the eigenvalues by employing a number of extra updates when the following conditions are satisfied:

$$b_k > 1, \qquad h_k < 1 \quad \text{and} \quad b_k + h_k < 2. \tag{25}$$

Otherwise, the standard SR1 update is employed.

4. Optimal updating under the σ-measure

Although the SR1 update possesses desirable features, it has some major drawbacks. The SR1 update is unstable in the sense that it may not preserve positive definiteness even when updated from a positive definite matrix, and the denominator can be zero or numerically zero in (10) in which case it could lead to numerical instabilities. In order to overcome these negative aspects, the SR1 update must be adjusted in some manner whenever it fails to exist or is not positive definite. Our most important aim in this section is to seek a matrix D such that the matrix updated from D preserves as much information as possible from the previous steps while maintaining positive definiteness of the update.

For this purpose, we should find a measure for deriving the optimal scaling factor for the SR1 update. Davidon [20] and Shanno and Phua [21] considered the κ-measure (l_2-condition number)

$$\kappa(P) = \frac{\zeta_{\max}}{\zeta_{\min}}, \tag{26}$$

where P is an $n \times n$ positive definite matrix, and $\zeta_{\max}$ and $\zeta_{\min}$ are the largest and the smallest eigenvalues of P.

Davidon [20] used the κ-measure (l_2-condition number) to choose an optimally conditioned update in the Broyden class whereas Shanno and Phua [21] used it to obtain the optimal scaling factor for the BFGS update. However, it was shown by Wolkowicz [22] that the κ-measure is not a good choice for finding an optimal scaling factor for the SR1 update.

For this reason, Dennis and Wolkowicz [10] suggested a measure (known as the σ-measure) based on the volume of the symmetric difference between the two ellipsoids corresponding to the most recent updates, say B_k and B_{k+1}.

The σ-measure is defined as follows:

$$\sigma(P) = \frac{\zeta_{\max}}{\det(P)^{1/n}}, \tag{27}$$

where "det" denotes the determinant, P is an $n \times n$ positive definite matrix and $\zeta_{\max}$ is the largest eigenvalue of P. Wolkowicz [22] showed that the σ-measure yields an optimally scaled SR1 update.

The following theorem, which is a straight result of Corollary 2.1 due to Leong and Hassan [11], shows that the "best" semi-positive definite (s.p.d.) SR1 updates for B_{k+1} under the σ-measure given by (27) are scaled, optimally conditioned, SR1 updates such as those in [23].

Theorem 1. *Suppose that*

$$\mu_k = \frac{s_k^T s_k}{s_k^T y_k} - \left(\frac{\left(s_k^T s_k\right)^2}{\left(s_k^T y_k\right)^2} - \frac{s_k^T s_k}{y_k^T y_k} \right)^{1/2}. \tag{28}$$

Then the inverse SR1 matrix updated from $\mu_k I$:

$$H_{k+1} = \mu_k I + \frac{(s_k - \mu_k y_k)(s_k - \mu_k y_k)^T}{y_k^T (s_k - \mu_k y_k)}, \tag{29}$$

is the unique solution of

$$\begin{aligned} &\min \sigma(H_{k+1}^{-1}), \\ &\text{s.t. } H_{k+1}^{-1} s_k = y_k, \qquad H_{k+1}^{-1} \text{ is s.p.d..} \end{aligned} \tag{30}$$

5. Description of the algorithm

Extra updates for the symmetric rank-one method (EXSR1)

Step 0. Given an initial point $x_0 \in \mathbb{R}^n$, and an initial positive definite matrix $H_0 = I$, compute $f(x_0)$ and $g_0 = \nabla f(x_0)$. Set $k = 0$.

Step 1. Termination test. If the convergence criterion $\|g_k\| \le \varepsilon$ is achieved, then stop.

Step 2. Compute a QN direction, d_k, by using $d_k = -H_k g_k$.

Step 3. Find an acceptable step length, α_k, such that the Wolfe conditions

$$f(x_k + \alpha_k d_k) \le f(x_k) + \delta_1 \alpha_k g_k^T d_k, \tag{31}$$

$$g(x_k + \alpha_k d_k)^T d_k \ge \delta_2 g_k^T d_k, \tag{32}$$

where $0 < \delta_1 < \delta_2 < 1$, $\delta_1 < \frac{1}{2}$, are satisfied.

Step 4. Set $x_{k+1} = x_k + \alpha_k d_k$.

Step 5. Calculate b_k and h_k defined by (21) and (22).

Step 6. Update H_k to H_{k+1} by using the SR1 formula (12).

Step 7. If condition (25) is satisfied, then replace the update H_{k+1} by $\bar{H}_{k+1}^{(1)}$.

Check the stabilizing options in Step 8, then compute $\bar{H}_{k+1}^{(1)}$ by using the EXSR1 formula

$$\bar{H}_{k+1}^{(l+1)} = \bar{H}_k^{(l)} + \frac{(\rho_l - \bar{H}_k^{(l)} \zeta_l)(\rho_l - \bar{H}_k^{(l)} \zeta_l)^T}{\zeta_l^T (\rho_l - \bar{H}_k^{(l)} \zeta_l)}, \quad l = 1, 2, \ldots, m, \tag{33}$$

where $m = 2$ and

$$\rho_l = s_{k-m+l}, \qquad \zeta_l = y_{k-m+l}, \quad 1 \le l \le m.$$

Use $H_{k+1} = \bar{H}_{k+1}^{(m+1)}$ to compute H_{k+1} and go to Step 9.

Step 8. Stabilizing. If

$$s_k^T y_k - y_k^T H_k y_k < 0, \quad \text{(the Hessian matrix may not be positive definite)}$$

or

$$\left|y_k^T (s_k - H_k y_k)\right| < t \, \|y_k\| \, \|s_k - H_k y_k\| ,$$

where $t \in (0, 1)$ (the denominator in the Hessian matrix is sufficiently close to zero) or

$$\|H_k\|_\infty > L, \quad \text{(where } L = 10^{10} \text{ is used)},$$

set $H_{k+1} = \mu_k I$, where $\mu_k = \frac{s_k^T s_k}{y_k^T s_k} - \left\{ \frac{\left(s_k^T s_k\right)^2}{\left(y_k^T s_k\right)^2} - \frac{s_k^T s_k}{y_k^T y_k} \right\}^{1/2}$.

Step 9. Set $k := k + 1$, and go to Step 1.

Note that since employing extra updates is not necessary at every iteration, we consider employing them only when the conditions in (25) hold.

6. Implementation and numerical experiments

In this section, we report the numerical comparisons of the proposed EXSR1 method, the "extra BFGS" (EXBFGS) method from [17] and the NSSR1 method (the SR1 method that restarts with the identity matrix). Some classical test functions with standard starting points are selected for testing our algorithms on a variety of 62 unconstrained optimization test problems, where the list of selected problems is given in Table 1. Each function is tested with dimensions varied from 2 to 1000 variables. The set of test functions used in the numerical experiments is fundamentally described in Bongartz et al. [24] along with other standard optimization test problems from [25] and [26].

Table 1
Selected test problems for comparing EXSR1 with the EXBFGS and NSSR1 methods.

Ex.Freudenstein & Roth,Ex.Trigonometric,Ex.Rosenbrock,Ex.White&Holst,
Ex.Beale, Ex.Penalty, Perturbed Quadratic, Raydan 1–2, Diagonal 1–5,
Hager, Ge.Tridiagonal 1, Ex.Tridiagonal 1–2, Ex.Three Expo Terms,
Ge.Tridiagonal 2, Ex.Himmelblau, Ge.PSC1, SQ1, SQ2,
Ex.PSC1, Ex.Powell, Ex.BD1, Ex.DENSCHNF,
Ex.Maratos, Ex.Cliff, QDP, Ex.Wood, STAIRCASE S1, LIARWHD,
Ex.Hiebert, QF1, Ex.QP1, Ex.QP2, QF2, COSINE, TPQ,
BDQRTIC, TRIDIA, ARWHEAD, NONDIA, NONDQUAR, DQDRTIC,
EG2, DIXMAANA, DIXMAANB, DIXMAANC, DIXMAANE,
PPQ1, Broyden Tridiagonal, Almost Perturbed Quadratic,
DIXON3DQ, DIXMAANF, DIXMAANG, DIXMAANH,
FLETCHCR, DIXMAANI, DIXMAANJ, DIXMAANK,

All algorithms use exactly the same implementation of the Wolfe line-search conditions (31) and (32) with $\delta_1 = 10^{-4}$ and $\delta_2 = 0.9$. The codes are written in Fortran 77 and in double-precision arithmetic. All tests are implemented on a PC with a 1.8 GHz Pentium IV processor and 1 GB of RAM. The step length $\alpha_k = 1$ is always tried first. Once the number of iterations or function evaluations exceeds 1000, we stop the algorithm.

As regards the stopping criterion to be used in our numerical experiments, in all algorithms, convergence is assumed if

$$\|g_k\| \leq \varepsilon, \quad \text{where } \varepsilon = 10^{-5}.$$

A total of 942 runs were performed. The new approach can solve over 95% of the test problems, where EXBFGS can solve 93% and NSSR1 solves 85%.

The dimensions of the test functions were classified into those of "small" ($2 \leq n \leq 100$), "medium" ($101 \leq n \leq 500$) and "large" ($501 \leq n \leq 1000$) size.

Since a large set of problems is used, we give summaries of the results from these experiments in Tables 2–5, by presenting the ratios of the geometric and arithmetic means of the number of iterations (Itrn.) and the function/gradient evaluations (Feval.) required to solve all the problems in the given test set. The results are all shown in Tables 2–5.

Table 2
Relative efficiency of EXSR1, EXBFGS and NSSR1 for small dimensions ($2 \leq n \leq 100$).

Method	EXSR1		EXBFGS		NSSR1	
	Itrn.	Feval.	Itrn.	Feval.	Itrn.	Feval.
Arithmetic mean	1	1	1.05	1.08	1.12	1.16
Geometric mean	1	1	1.01	1.04	1.07	1.11

Table 3
Relative efficiency of EXSR1, EXBFGS and NSSR1 for medium dimensions ($101 \leq n \leq 500$).

Method	EXSR1		EXBFGS		NSSR1	
	Itrn.	Feval.	Itrn.	Feval.	Itrn.	Feval.
Arithmetic mean	1	1	1.09	1.13	1.18	1.20
Geometric mean	1	1	1.04	1.08	1.14	1.16

Table 4
Relative efficiency of EXSR1, EXBFGS and NSSR1 for larger dimensions ($501 \leq n \leq 1000$).

Method	EXSR1		EXBFGS		NSSR1	
	Itrn.	Feval.	Itrn.	Feval.	Itrn.	Feval.
Arithmetic mean	1	1	1.15	1.19	1.22	1.25
Geometric mean	1	1	1.10	1.14	1.17	1.20

Table 5
Overall summary (942 runs).

Method	EXSR1		EXBFGS		NSSR1	
	Itrn.	Feval.	Itrn.	Feval.	Itrn.	Feval.
Arithmetic mean	1	1	1.10	1.14	1.17	1.20
Geometric mean	1	1	1.05	1.09	1.13	1.16

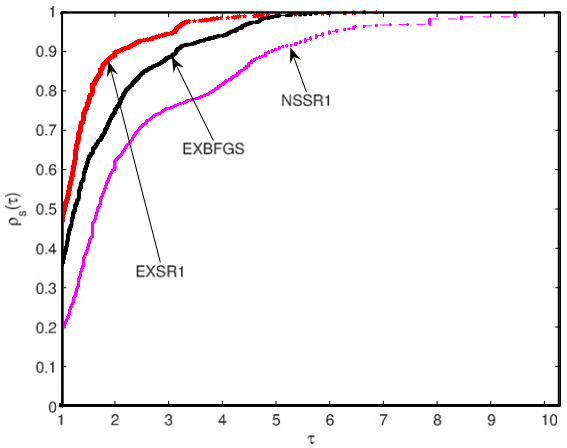

Fig. 1. Performance profiles of EXSR1, EXBFGS and NSSR1 based on iterations.

From Tables 2–5, it can be observed that EXSR1 performs clearly better than EXBFGS and NSSR1 in several cases. EXSR1 seems to perform better than EXBFGS for the most difficult problems, although the improvement is seen more in respect of iterations than function/gradient evaluations. Indeed the EXSR1 algorithm has the desirable feature that the extra updates improve the performance of the algorithms in most cases.

According to the Table 5, EXSR1 defines the most efficient numerical performance in terms of means, in comparison with the other algorithms. The improvement of EXSR1 over EXBFGS lies in EXSR1 needing 5% and 9% fewer, on average, iteration calls and function/gradient evaluations respectively. Similarly, the improvement of EXSR1 over NSSR1 lies in NSSR1 needing 13% and 16% more, on average, iteration calls and function/gradient evaluations, respectively, than STSR1. The STSR1 algorithm is 5% to 20%, on average, cheaper than the EXBFGS and NSSR1 methods. This observation shows a significant improvement of the EXSR1 method over the EXBFGS and NSSR1 methods as the dimension of the problem increases.

Overall, EXSR1 requires 13 185 (total number of) iterations and 16 657 (total number of) function/gradient evaluations for solving 62 test problems, while EXBFGS requires 15 672 and 19 710 and NSSR1 requires 23 059 and 27 937.

Therefore, since the numbers of iterations and function/gradient evaluations needed by the EXSR1 method are fewer than those for EXBFGS and NSSR1, EXSR1 seems to be a better option for optimizing the selected problems.

The results are also presented using the performance profile tool proposed by Dolan and Moré [27], comparing the performances of a number of n_s solvers running on the set of n_p problems.

We have set $n_s = 3$ and $n_p = 62$ in the cases of the problems listed in Table 1. The corresponding performance profiles are plotted in Figs. 1–3.

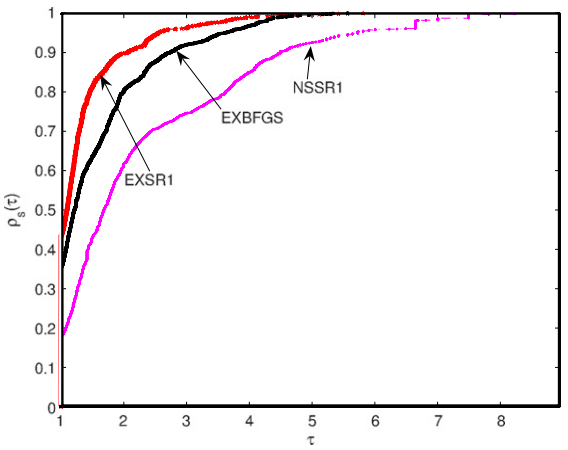

Fig. 2. Performance profiles of EXSR1, EXBFGS and NSSR1 based on function/gradient calls.

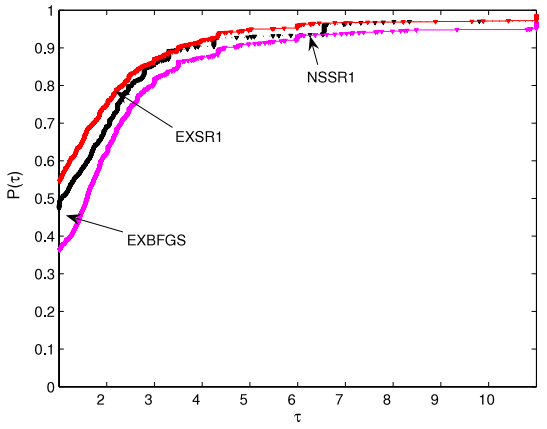

Fig. 3. Performance profiles of EXSR1, EXBFGS and NSSR1 based on CPU time.

According to the Figs. 1 and 2, it is not difficult to conclude that EXSR1 outperforms the EXBFGS method. Meanwhile the improvement of the performance of the EXSR1 method over that of the NSSR1 counterpart is significant. From Fig. 3, it can be observed that as regards the CPU time metric, the proposed method is fastest, followed by the EXBFGS and the NSSR1 methods.

7. Conclusion

A technique for improving updated Hessian approximations for use in a symmetric rank-one (SR1) method has been introduced. Our strategy is to employ further updates to the SR1 Hessian at certain iterations. As regards the drawbacks of the SR1 method, a simple safeguard strategy is designed for the EXSR1 algorithm to deal with the non-positive definite updates and to prevent the possible breakdown of the denominator. The method has been compared experimentally with the NSSR1 method and with the "extra BFGS" method originally introduced in [17]. From the numerical results we can conclude that the proposed method is encouraging, as compared with the EXBFGS and NSSR1 methods. In particular we observe that EXSR1 requires, quite frequently, fewer iterations, and hence fewer function/gradient evaluations, than EXBFGS and NSSR1 for a set of test problems. In many cases, EXSR1 solves the problem, and this is clearly an advantage of the proposed method.

References

[1] R. Fletcher, Practical Methods of Optimization, John Wiley & Sons, New York, 1987.
[2] H.F.H. Khalfan, Topics in quasi-Newton methods for unconstrained optimization, Ph.D. Thesis, University of Colorado, Colorado, 1989.
[3] H.F.H. Khalfan, R.H. Byrd, R.B. Schnabel, A theoretical and experimental study of the symmetric rank one update, SIAM J. Optimum 3 (1993) 1–24.
[4] A.R. Conn, N.I.M. Gould, Ph. Toint, Convergence of quasi-Newton matrices generated by the symmetric rank one update, Math. Program. 48 (1991) 549–560.
[5] H.F.H. Khalfan, R.B. Schnabel, Analysis of a symmetric rank-one trust region method, Tech. Rep. CU-CS-657-93, Department of Computer Science, University of Colorado at Boulder, USA, 1993.
[6] L.P. Sun, Computational experience with rank-one positive definite quasi-Newton algorithms, Optimum 16 (1984) 673–681.

[7] M.R. Osborne, L.P. Sun, A new approach to the symmetric rank one updating algorithm, IMA J. Numer. Anal. 19 (1999) 497–507.
[8] P.K.H. Phua, Eigenvalues and switching algorithms for quasi-Newton updates, Optimum 42 (1997) 185–217.
[9] P.K.H. Phua, Multi-directional parallel algorithms for unconstrained optimization, Optimum 38 (1996) 107–125.
[10] J.E. Dennis, H. Wolkowicz, Sizing and least change secant methods, SIAM J. Numer. Anal. 30 (5) (1993) 1291–1313.
[11] W.J. Leong, M.A. Hassan, A restarting approach for the symmetric rank one update for unconstrained optimization, Comput. Optim. Appl. 42 (3) (2009) 327–334.
[12] F. Modarres, M.A. Hassan, W.J. Leong, Improved Hessian approximation with modified secant equations for symmetric rank-one method, J. Comput. Appl. Math. 235 (2011) 2423–2431.
[13] R.H. Byrd, D.C. Liu, J. Nocedal, On the behaviour of Broyden's class of quasi-Newton methods, SIAM J. Optim. 2 (1992) 533–557.
[14] M.J.D. Powell, How bad are the BFGS and DFP methods when the objective function is quadratic? Math. Program. 34 (1986) 34–47.
[15] R. Fletcher, An Overview of Unconstrained Optimization. Tech. Report NAI149, Department of Mathematics and Computer Science, Dundee University, 1993.
[16] J. Nocedal, Theory of algorithms for unconstrained optimization, Acta Numer. 1 (1992) 199–242.
[17] M. Al-Baali, Extra updates for the BFGS method, Optim. Methods Softw. 13 (2000) 159–179.
[18] M. Al-Baali, Improved Limited Memory BFGS Method for Large Scale Optimization. Research Report DOMAS 9811, Dept. of Mathematics and Statistics, Sultan Qaboos University, Sultanate of Oman, 1998.
[19] M. Al-Baali, A. Fuduli, R. Musmanno, On the performance of switching BFGS/SR1 algorithms for unconstrained optimization, Optim. Methods Softw. 19 (2004) 153–164.
[20] W.C. Davidon, Optimally conditioned optimization algorithms without line search, Math. Program. 9 (1975) 1–30.
[21] D.F. Shanno, K.H. Phua, Matrix conditioning and nonlinear optimization, Math. Program. 14 (1978) 149–160.
[22] H. Wolkowicz, Measure for symmetric rank-one updates, Math. Oper. Res. 19 (4) (1994) 815–830.
[23] C.M. Ip, M.J. Todd, Optimal conditioning and convergence in rank one quasi-Newton updates, SIAM J. Numer. Anal. 25 (1988) 206–221.
[24] I. Bongartz, A.R. Conn, N.I.M. Gould, Ph.L. Toint, CUTE: constrained and unconstrained testing environment, ACM Trans. Math. Softw. 21 (1995) 123–160.
[25] N. Andrei, An unconstrained optimization test functions collection, Adv. Model. Optim. 10 (2008) 147–161.
[26] J.J. Moré, B.S. Garbow, K.E. Hillstorm, Testing Unconstrained optimization software, ACM Trans. Math. Softw. 7 (1) (1981) 17–41.
[27] E.D. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, Math. Program. 91 (2) (2002) 201–203.